

(19)



JAPANESE PATENT OFFICE

BEST AVAILABLE COPY

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **07123013 A**(43) Date of publication of application: **12 . 05 . 95**

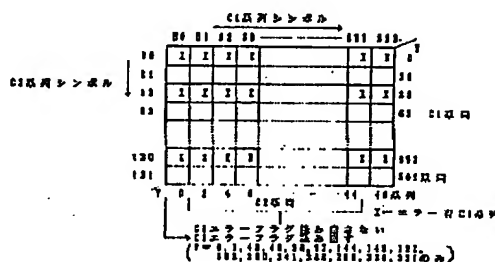
(51) Int. Cl.

H03M 13/00**G11B 20/18****G11B 20/18****G11B 20/18**(21) Application number: **06082505**(22) Date of filing: **29 . 03 . 94**(30) Priority: **30 . 08 . 93 JP 05237387**(71) Applicant: **VICTOR CO OF JAPAN LTD**(72) Inventor:
MIYASHITA IZUMI
UENO SHOJI
UMEZONO TAKU**(54) ERROR CORRECTION DEVICE****(57) Abstract:**

PURPOSE: To allow the error correction device for a digital signal to correct errors in the signal at a high speed by adopting duplicate Reed Solomon codes.

CONSTITUTION: A C1 series is made of 24-words×512-blocks (W) and C2 series is made of 32-words×384-blocks (V) in one frame of main data of a DCC. The 384-blocks (V) of the C2 series are grouped into 16 groups so that words of the C2 series are in duplicate with those of the C1 series. A C1 error flag is read from the blocks 1, 48, 49, 96, 97, 144, 145, 192, 193, 240, 241, 288, 289, 336, 337 and evaluated, its error location and error numbers to be counted are stored, but the C1 error is not read from the other blocks. In the other mode, whether or not the correction of the C2 series is to be executed is discriminated depending on the value of the C1 flag in erroneous data based on the location of the erroneous data of the C2 series.

COPYRIGHT: (C)1995,JPO



THIS PAGE BLANK (USPTO)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-123013

(43) 公開日 平成7年(1995)5月12日

| (51) Int. Cl. ⁶ | 識別記号 | 庁内整理番号 | F I | 技術表示箇所 |
|----------------------------|---------|----------|-----|--------|
| H 0 3 M 13/00 | | 8730-5 J | | |
| G 1 1 B 20/18 | 5 3 2 E | 9074-5D | | |
| | 5 3 6 E | 9074-5D | | |
| | 5 7 2 B | 9074-5D | | |
| | G | 9074-5D | | |

審査請求 未請求 請求項の数 2 F D (全 21 頁)

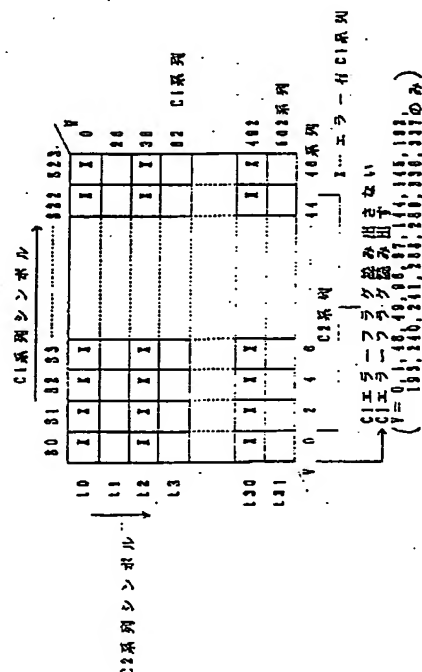
| | | | |
|--------------|-----------------|----------|--|
| (21) 出願番号 | 特願平6-82505 | (71) 出願人 | 000004329 日本ビクター株式会社 神奈川県横浜市神奈川区守屋町3丁目12番地 |
| (22) 出願日 | 平成6年(1994)3月29日 | (72) 発明者 | 宮下 泉 神奈川県横浜市神奈川区守屋町3丁目12番地 日本ビクター株式会社内 |
| (31) 優先権主張番号 | 特願平5-237387 | (72) 発明者 | 植野 昭治 神奈川県横浜市神奈川区守屋町3丁目12番地 日本ビクター株式会社内 |
| (32) 優先日 | 平5(1993)8月30日 | (72) 発明者 | 梅園 卓 神奈川県横浜市神奈川区守屋町3丁目12番地 日本ビクター株式会社内 |
| (33) 優先権主張国 | 日本 (J P) | (74) 代理人 | 弁理士 二瓶 正敬 |

(54) 【発明の名称】 エラー訂正装置

(57) 【要約】

【目的】 デジタル信号のエラー訂正装置にあって2重のリードソロモン符号を用いて高速でエラー訂正する。

【構成】 DCCのメインデータの1フレームはC1系列が24ワード×512ブロック(W)で構成され、C2系列が32ワード×384ブロック(V)で構成されている。C1系列のワードが重複するようにC2系列の384ブロック(V)が16個にグループ分けされ、V=1, 48, 49, 96, 97, 144, 145, 192, 193, 240, 241, 288, 289, 336, 337のブロックではC1エラーフラグを読み出してC1フラグ評価を行い、そのエラーの位置と数をカウントして記憶するが、他のブロックではC1エラーフラグを読み出さない。他の態様ではC2系列の誤りデータの位置を基に、その誤りデータのC1フラグの値によってC2系列の訂正動作を実行するか否かを判断している。



【特許請求の範囲】

【請求項1】 C1系列とC2系列の2重のリードソロモン符号を採用したエラー訂正装置であって、前記C1系列の各ブロックのエラーを訂正するとともに、エラーが発生したブロック毎にエラーフラグを記憶するC1系列エラー訂正手段と、前記C1系列のワードと前記C2系列のワードの巡回性に基づいて前記C2系列のブロックを前記C1系列のワードが重複するグループにグループ分けを行い、前記C1系列エラー訂正手段により記憶されて前記C2系列の各ワードに対応するエラーフラグを前記グループ毎に1回読み出して前記C2系列の各ブロックのエラーを訂正するC2系列エラー訂正手段とを有するエラー訂正装置。

【請求項2】 C1系列とC2系列の2重のリードソロモン符号を採用したエラー訂正装置であって、前記C1系列の各ブロックのエラーを訂正するとともに、エラーが発生したブロック毎にエラーフラグを記憶するC1系列エラー訂正手段と、前記C1系列エラー訂正手段により記憶されて前記C2系列の各ワードに対応するエラーフラグを読み出して前記C2系列の各ブロックのエラーを訂正するC2系列エラー訂正手段とを有し、前記C2系列エラー訂正手段が、前記C2系列の訂正演算により得られた誤りデータの位置を基に、その誤りデータのC1フラグを読み込み、そのC1系列のエラーフラグの値によって前記C2系列の訂正動作を実行するか否かを判断する手段を有するエラー訂正装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、2重のリードソロモン符号を採用したエラー訂正装置に関し、特にDCC（デジタルコンパクトカセット）に記録されたオーディオ信号を再生する場合に好適なエラー訂正装置に関する。

【0002】

【従来の技術】一般に、DCCでは、主としてランダムエラー訂正用のC1系列と主としてバーストエラー訂正用のC2系列の2重のリードソロモン（RS）符号がクロスインターリーブされている。例えばDCCでは、メインデータの1フレームを12288ワード（1ワード＝8ビット）で構成し、このデータを2つの法則（C1、C2インターリーブ）に基づいて1系列（ブロック）当たり24ワードのC1系列と、1ブロック当たり32ワードのC2系列の2つの系列に分けて訂正を行う。すなわち、DCCでは12288個のデータをC1とC2の両方の系列で分類している。

【0003】DCCシステムの再生時のエラー訂正ではC1系列の訂正を最初に行う。C1系列は1ブロックが24ワードのデータs0～s23により構成され、1フレームが512（12288÷24）のブロックW（＝

0～511）により構成されている。C1訂正では1ブロックについてエラーが0ワード、1ワード、2ワード、3ワード以上の4種類に分けて訂正を行い、その結果に応じて表1に示すようにエラーフラグとしてそれぞれ「00」、「01」、「03」、「07」の各8ビットデータをメモリに書き込む。したがって、C1のエラーフラグデータは、1つのブロックに1つであるので、1フレーム当たり合計512個がメモリに書き込まれる。

【0004】

【表1】

C1エラーフラグ

| | F0 | F1 | F2 |
|--------|----|----|----|
| エラーなし | 0 | 0 | 0 |
| 1ワード訂正 | 1 | 0 | 0 |
| 2ワード訂正 | 1 | 1 | 0 |
| 訂正不可 | 1 | 1 | 1 |

【0005】上記C1系列の最終ブロック（W＝511）の訂正が終了するとC2系列の訂正を開始する。C2系列は1ブロックが32ワードのデータt0～t31により構成され、1フレームが384（12288÷32）のブロックV（＝0～383）により構成される。また、C2訂正では上記C1訂正によりエラー位置が判っている場合、すなわち、C2系列における32ワードデータt0～t31の内、どのデータがエラーであるかが判っている場合にエラーを訂正する、いわゆるイレージャ訂正を行う。

【0006】ここで、C2系列の32ワードデータt0～t31はC1系列においてもいずれかのブロック（W＝0～511）に属しているので、イレージャ訂正ではC2系列のデータについてそのデータが属しているC1系列のブロックWのエラーフラグデータを読み出し、そのC1ブロックが2ワード訂正（エラーフラグ「03」）、或いは3ワード以上のエラー（エラーフラグ「07」）である場合にそのC2データがエラーであるとしてイレージャ訂正を行う。

【0007】従来、この種のエラー訂正装置では、C2系列の訂正を行う場合に384個の全てのブロックVに関してC1系列の訂正におけるエラーフラグを読み出して評価するように構成されている。すなわち、従来のDCCシステムのイレージャ訂正では、1フレーム当たりエラーフラグを12288回（384ブロック×32ワード）読み出している。

【0008】

【発明が解決しようとする課題】しかしながら、上記従来のエラー訂正装置では、C2系列の訂正を行う場合に全てのブロックVに関してC1系列の訂正におけるエラーフラグを評価するので、アドレッシングに時間がかかり、高速化を妨げるという第1の問題点がある。また、

C 2 系列の 1 個又は 2 個の誤りを検出するに際し、可能性として、例えば実際は 3 個ある誤りを 1 個の誤りと誤認し、あるいは例えば実際は 4 個ある誤りを 2 個の誤りと誤認することもありえる。以下これを第 2 の問題点という。

【0009】本発明は上記従来の第 1 の問題点に鑑み、2 重のリードソロモン符号を用いて高速でエラー訂正することができるエラー訂正装置を提供することを目的とする。さらに本発明は、上記従来の第 2 の問題点に鑑み、誤り個数の誤認を防止し、訂正誤りの発生確率を大幅に低下させることができるエラー訂正装置を提供することを目的とする。

【0010】

【課題を解決するための手段】本発明は第 1 の上記目的を達成するために、C 1 系列のワードと C 2 系列のワードの巡回性に基づいて C 2 系列のブロックを C 1 系列のワードが重複するグループにグループ分けし、重複したエラーフラグは読み出さないようにしている。すなわち本発明によれば、C 1 系列と C 2 系列の 2 重のリードソロモン符号を採用したエラー訂正装置であって、前記 C 1 系列の各ブロックのエラーを訂正するとともに、エラーが発生したブロック毎にエラーフラグを記憶する C 1 系列エラー訂正手段と、前記 C 1 系列のワードと前記 C 2 系列のワードの巡回性に基づいて前記 C 2 系列のブロックを前記 C 1 系列のワードが重複するグループにグループ分けを行い、前記 C 1 系列エラー訂正手段により記憶されて前記 C 2 系列の各ワードに対応するエラーフラグを前記グループ毎に 1 回読み出して前記 C 2 系列の各ブロックのエラーを訂正する C 2 系列エラー訂正手段とを有するエラー訂正装置が提供される。さらに、本発明によれば上記第 2 の目的を達成するために C 1 系列と C 2 系列の 2 重のリードソロモン符号を採用したエラー訂正装置であって、前記 C 1 系列の各ブロックのエラーを訂正するとともに、エラーが発生したブロック毎にエラーフラグを記憶する C 1 系列エラー訂正手段と、前記 C 1 系列エラー訂正手段により記憶されて前記 C 2 系列の各ワードに対応するエラーフラグを読み出して前記 C 2 系列の各ブロックのエラーを訂正する C 2 系列エラー訂正手段とを有し、前記 C 2 系列エラー訂正手段が、前記 C 2 系列の訂正演算により得られた誤りデータの位置を基に、その誤りデータの C 1 フラグを読み込み、その C 1 系列のエラーフラグの値によって前記 C 2 系列の訂正動作を実行するか否かを判断する手段を有するエラー訂正装置が提供される。

【0011】

【作用】本発明では、1 フレームのデータが s ワード×W ブロックで構成された C 1 系列と、1 フレームのデータが t ワード×V ブロックで構成された C 2 系列の 2 重のリードソロモン符号を用いて C 2 系列をエラー訂正する場合に、C 1 系列のワードと C 2 系列のワードの巡回

性に基づいて C 2 系列のブロックを C 1 系列のワードが重複するグループにグループ分けすることにより C 1 系列のエラー訂正時のエラーフラグがグループ毎に 1 回読み出される。したがって、グループ分けされた C 2 系列の他のブロックでは C 1 系列のエラー訂正時のエラーフラグが読み出されないので、高速でエラー訂正することができる。

【0012】

【実施例】以下、図面を参照して本発明の実施例について説明する。図 1 は本発明に係るエラー訂正装置の一実施例におけるエラー訂正方法を示す説明図、図 2 は C 1 系列のワードと C 2 系列のワードの関係を示す説明図、図 3 は C 2 系列の第 0 ～第 95 系列と C 1 系列の関係を示す説明図、図 4 は C 2 系列の第 96 ～第 191 系列と C 1 系列の関係を示す説明図である。

【0013】まず、図 2 ～図 4 を参照して C 1 系列と C 2 系列の各ワードの関係を説明する。前述したように DCC システムのメインデータの C 1、C 2 系列はそれぞれ、

C 1 : GF (2⁸) Reed Solomon Code (24, 20, 5)

C 2 : GF (2⁸) Reed Solomon Code (32, 26, 7)

のコード体系で構成されている。また、1 フレームは 12288 ワード (1 ワード=8 ビット) で構成されるとともに、C 1、C 2 系列の 1 フレームはそれぞれ C 1 : W=0～511 の 512 系列 (ブロック)、C 2 : V=0～383 の 384 系列 (ブロック) で構成されている。

【0014】そして、C 1 系列と C 2 系列の各ワードの関係は、図 2 に示すように C 2 系列の

(1) V=0, 2, 4～46

の 24 個の系列の各ワードデータ (t=0～31) を先頭シンボル (t=0) から順次縦方向に配列するとともに、各 C 2 系列の同一シンボル番号 (t=0～31) の各ワードデータを V=0, 2, 4～46 の系列順に横方向に配列すると、この C 2 系列の横方向の各ワードデータは、C 1 系列の W=0, 26, 36, 62～492、502 の各ワードデータ (s 0～s 23) と同一となる。

【0015】すなわち、例えば C 2 シンボル t=0 の横方向の各ワードデータは C 1 系列の W=0 の各ワードデータ s 0～s 23 と同一であり、C 2 シンボル t=1 の横方向の各ワードデータは C 1 系列の W=26 の各ワードデータ s 0～s 23 と同一であり、C 2 シンボル t=3 の横方向の各ワードデータは C 1 系列の W=62 の各ワードデータ s 0～s 23 と同一である。更に説明すると、C 2 系列における第 0 系列 (V=0) の第 0 シンボル (t=0) のワードデータは C 1 系列における W=0 のワードデータ s 0 であり、また、C 2 系列における第

5

4系列 ($V=4$) の第2シンボル ($t=2$) のワードデータはC1系列における $W=36$ のワードデータs2である。

【0016】同様に、図3に示すようにC2系列の

(2) $V=1, 3, 5 \sim 45, 47$

(3) $V=48, 50 \sim 92, 94$

(4) $V=49, 51 \sim 93, 95$

また、図4に示すようにC2系列の

(5) $V=96, 98 \sim 140, 142$

(6) $V=97, 99 \sim 141, 143$

(7) $V=144, 146 \sim 188, 190$

(8) $V=145, 147 \sim 189, 191$

のそれぞれ24個の系列の各ワードデータ ($t=0 \sim 31$) を図2と同様に先頭シンボル ($t=0$) から順次縦方向に配列するとともに、各C2系列の同一シンボル番号 ($t=0 \sim 31$) の各ワードデータを系列順に横方向に配列すると、このC2系列の横方向の各ワードデータは図3および図4に示す番号WのC1系列の各ワードデータ ($s0 \sim s23$) と同一となる。

【0017】また、図は省略されているが、C2系列の

(9) $V=192, 194 \sim 236, 238$

(10) $V=193, 195 \sim 237, 239$

(11) $V=240, 242 \sim 284, 286$

(12) $V=241, 243 \sim 285, 287$

(13) $V=288, 290 \sim 332, 334$

(14) $V=289, 291 \sim 333, 335$

(15) $V=336, 338 \sim 380, 382$

(16) $V=337, 339 \sim 381, 383$

の場合にも同様に、このC2系列の横方向の各ワードデータは他のC1系列の各ワードデータ ($s0 \sim s23$) と同一となる。

【0018】すなわち、DCCシステムは、C1系列の1フレームが $W=0 \sim 511$ の512系列で構成され、C2系列の1フレームが $V=0 \sim 383$ の384系列で構成されているので、C1系列のワードとC2系列のワードの巡回性に基づいてC2系列のブロックVをC1系列のワードが重複する24系列毎にまとめると、16

($=384 \div 16$) 通りにグループ分けすることができる (この明細書では、上記ブロックと区別するために「グループ」を用いる)。なお、言うまでもなく、 $V=0 \sim 383$ の各系列は $t=0 \sim 31$ の各データワードを有するので、1フレームでは $32 \times 16 = 512$ 通りのC1系列となり、C1系列の1フレーム分となる。

【0019】ここで、後述するようにC1系列の訂正では、エラーを検出して訂正を実際に行った場合、或いは3ワード以上のエラーを検出して訂正できなかった場合にそのワードにエラー有りを示すフラグを与えてメモリに記憶し、続くC2系列の訂正では、C1系列の訂正においてエラーがあったC1系列のワードに含まれるデータは全てエラーであるとみなして訂正 (イレージャ訂

6

正) を行うが、このC2系列のイレージャ訂正では、訂正を行う前にC1系列の訂正における訂正結果をフラグとして読み出すことにより評価するいわゆるC1フラグ評価が行われる。

【0020】例えば図1において「×」で示すようにC1系列の $W=0, 36, 492$ の系列にエラーがあった場合、C2系列の $V=0, 2 \sim 44, 46$ の各系列では共に、32ワードのうちシンボル $t=0, 2, 30$ の3ワードがエラーであるとみなしてイレージャ訂正を行う。なお、前述したように例えばC1系列の訂正において $W=36$ の系列に3ワード以上のエラーがあった場合にはエラーフラグ「07」が記憶される。

【0021】そこで、本実施例では、 $V=0, 2 \sim 4, 46$ の第1グループでは各系列のイレージャ訂正を行う際に読み出すエラーフラグは全て読み出さず、 $V=0$ の系列のみのC1エラーフラグを読み出してC1フラグ評価を行い、そのエラーの位置と数をカウントして記憶することにより、続く $V=2 \sim 44, 46$ の各系列のイレージャ訂正ではエラーフラグを読み出す必要がないようにしている。

【0022】また、上記第2～第16グループの各イレージャ訂正においても同様に、それぞれ $V=1, 48, 49, 96, 97, 144, 145, 192, 193, 240, 241, 288, 289, 336, 337$ の系列のみのエラーフラグを読み出してC1フラグ評価を行い、そのエラーの位置と数をカウントして記憶することにより、同一グループでは他のC1エラーフラグを読み出すことなく、全てのC2系列のイレージャ訂正を行うことができる。したがって、本実施例によれば、1フレームのC1フラグ評価は、1グループ毎に1回の合計16回でよい。

【0023】次に、上記エラー訂正方法を実現する装置について説明する。図5は本発明に係る誤り訂正装置の一実施例を示すブロック図、図6はリードソロモン符号のC1系列を訂正するルーチンを説明するためのフローチャート、図7はリードソロモン符号のC2系列を訂正するルーチンを説明するためのフローチャート、図8は図7のC2系列訂正ルーチンのイレージャルーチンを説明するためのフローチャート、図9はシンドローム修正ルーチンを説明するためのフローチャートである。なお、図1～図4において説明したC1エラーフラグ評価は、図5に示すフラグロケーション設定回路2により、また、図7に示すC2系列訂正ルーチンのステップ124において実行される。

【0024】先ず、図5に示す回路を概略的に説明すると、回路2～20が入力端子1から入力する信号の誤りを訂正するように構成され、また、この誤り訂正回路を構成する部分2～20は、回路2～8を含むシンドローム演算ブロック22と、回路9～14を含むラッチブロック23及び回路15～20を含む訂正ブロック26に

より大別される。また、このブロック22、23および26はアドレスブロック24と、インストラクションブロック25と補助トラック割り込みブロック27により制御され、特に割り込みブロック27はブロック22、23および26がDCCのメイントラックのC1、C2系列と補助トラックのC1系列を選択的に訂正するように割り込み処理を行う。

【0025】フラグロケーション設定回路2は、再生時にイレージャ訂正のためのC1エラーフラグ評価を行う回路であり、C2の2.4系列に1回C1エラーフラグを読み出し、C1の2ワードエラーおよび3ワードエラーの位置と数を検出する。また、この回路2はエラーフラグの読み出しのためのRAMアドレスを生成し、出力する。

【0026】パリティロケーション設定回路3は、記録時にイレージャ訂正を使ってパリティを計算するためにパリティの位置をメイントラック系列C1、C2と補助トラック系列AUXC1の各系列に合わせてロケーション選択回路4に出力する。ロケーション選択回路4は、再生モード時にフラグロケーション設定回路2からのエラーの位置を選択し、記録モード時にパリティロケーション設定回路3からのパリティの位置を選択し、レジスタ出力選択回路9、11に出力する。

【0027】シンドロームチェック回路5は、入力端子1より入力されるRAM(図示省略)からのデータを受け取って後述するようにC1系列では4個のシンドロームS0～S3を演算し、C2系列では6個のシンドロームS0～S5を演算してシンドローム選択回路6に出力する。シンドローム選択回路6はシンドローム選択回路6からのシンドローム、レジスタ19または16からの出力を選択し、指数に変換するためのテーブルの $\alpha-i$ 変換ROM7に出力する。

【0028】シンドローム格納レジスタ8には、 $\alpha-i$ 変換ROM7により $\alpha-i$ 変換されたシンドロームが格納され、このレジスタ8はまた、格納されたシンドロームが全て「00」の場合にフラグ「1」を出力する。レジスタ出力選択回路9、11は、シンドローム格納レジスタ8に格納されたシンドローム、ロケーション選択回路4により選択されたデータ及びレジスタ14に格納されたデータを選択して加算回路12に出力し、また、訂正データシンボルアドレスラッチ回路10は、エラー訂正を実行する時に演算により得られたエラー位置を記憶し、RAMアドレス出力回路24に出力する。

【0029】加算回路12はレジスタ出力選択回路9、11により選択されたデータを加算するが、この加算は α の指数部の加算であるので、インストラクションでは乗算となる。レジスタ入出選択回路13は、加算回路12の出力またはROM7により $\alpha-i$ 変換されたシンドロームを選択して後段のレジスタ14に記憶させる。なお、このレジスタ14は演算途中のデータを一時記憶し

てレジスタ出力選択回路9、11に出力するために用いられる。

【0030】 $i-\alpha$ 変換回路15は加算回路12の出力を $i-\alpha$ 変換し、このデータは排他的論理和回路17により、レジスタ19に格納されたデータと加算されて再度レジスタ19に格納される。レジスタ16は、2ワード訂正時に $Z^2 + Z + X = 0$ の解「Z」を求めるために「X」の値を入力すると「Z」の値に変換して記憶して出力する。訂正データ出力回路20は、 $i-\alpha$ 変換回路15からのデータとデータバス上の誤りデータとから排他的論理和回路18により得られる訂正データをデータバス上に出力する。

【0031】RAMアドレス出力回路24は、メインデータC1系列のRAMアドレス(C1RAMAD)と、メインデータC1系列のエラーフラグのRAMアドレス(C1FLGAD)と、メインデータC2系列のRAMアドレス(C2RAMAD)と、メインデータC1系列のエラーフラグのRAMアドレス(C2FLGAD)と、各系列のエラーフラグデータ(ERFLGBUF)を生成して出力する。

【0032】補助トラック割り込み検出回路27は、再生時にはAUX情報がメインデータであるC1、C2と同期していないので、割り込みの形式で演算、訂正を行い、再生時には入力信号の変化点を検出して割り込みフラグを出力する。なお、1フレーム中の入力信号の変化点は4回有り、1回について補助トラックデータが2系列ずつ処理される。補助トラック割り込み検出回路27はまた、補助トラックデータAUXC1系列のRAMアドレス(AXC1AD)と、補助トラックデータAUXC1系列のエラーフラグのRAMアドレス(AXFLGAD)を生成して出力する。

【0033】次に、インストラクション回路25を詳細に説明すると、まず、クロック発生器(CLOCKGEN)は各入力信号から、この装置内部で使用される各種クロックを生成する。インストラクションカウンタ(INSTCNT)はC1、C2、AUXC1のシンドローム演算及びC1エラーフラグ評価インストラクション用の10ビットカウンタであり、このカウンタの出力がインストラクションROM(INSTROM)25aのアドレスとなる。このインストラクションの1ステップは、クロックの立ち上がりから立ち上がりまでであり、クロックによりカウントアップする。また、このインストラクションのジャンプは、下記の飛び先アドレスをロードすることにより行われる。

【0034】インストラクションROM25aはインストラクションカウンタ(INSTCNT)から出力されるカウント値をアドレスとして16ビットデータを出力し、このデータがインストラクションの各ステップにおける処理動作を決定する。インストラクションセレクト(INSTSEL)はインストラクションROM25a

9

から出力される16ビットデータを処理の種類(シンドローム演算、エラーフラグ処理等)に応じて出力先を振り分け、この出力はクロックのタイミングで出力される。また、このセレクトはRAMアクセス時にインストラクションを止める信号を出力する。

【0035】ロードアドレス発生器(LOADAD)はインストラクションカウンタ(INSTCNT)から出力されるカウント値をラッチしたデータを読み取り、このデータがジャンプを行うアドレスの場合に各入力条件に従って飛び先アドレスを決定してインストラクション

カウンタ(INSTCNT)に出力する。
【0036】ここで、シンドローム演算と訂正処理のインストラクションが同時に進行しているが、RAMには同時にアクセスすることができないので、インストラクションコントローラ(INSTCONT)がアドレスを監視し、RAMアクセスが衝突しないようにインストラ

[C1]

$$S0 = W0 + W1 + W2 + \dots + W23$$

$$S1 = \alpha^{23}W0 + \alpha^{22}W1 + \alpha^{21}W2 + \dots + W23$$

$$S2 = \alpha^{46}W0 + \alpha^{44}W1 + \alpha^{42}W2 + \dots + W23$$

$$S3 = \alpha^{69}W0 + \alpha^{66}W1 + \alpha^{63}W2 + \dots + W23$$

[C2]

$$S0 = W0 + W1 + W2 + \dots + W31$$

$$S1 = \alpha^{31}W0 + \alpha^{30}W1 + \alpha^{29}W2 + \dots + W31$$

$$S2 = \alpha^{62}W0 + \alpha^{60}W1 + \alpha^{58}W2 + \dots + W31$$

$$S3 = \alpha^{93}W0 + \alpha^{90}W1 + \alpha^{87}W2 + \dots + W31$$

$$S4 = \alpha^{124}W0 + \alpha^{120}W1 + \alpha^{116}W2 + \dots + W31$$

$$S5 = \alpha^{155}W0 + \alpha^{150}W1 + \alpha^{145}W2 + \dots + W31$$

【0039】

【数2】

C1: S0 S1 S2 S3

C2: S0 S1 S2 S3 S4 S5

【0040】次いで、シンドロームS0～S3が全て「0」か否かを判別し(ステップ104)、YESの場合にはC1エラーフラグF0、F1、F2に共に「0」を書き込み(ステップ105)、次いでブロックアドレスを1つインクリメントし(ステップ106)、全ブロック(W=0～511)が終了しない場合にはステップ102に戻り、終了した場合には図7に示すC2訂正処理へ進む(ステップ107)。

【0041】他方、ステップ104においてシンドロームS0～S3が全て「0」でない場合には、先ず、次式(数3)に基づいて1ワードエラーを検出するための変形シンドローム $\sigma_1 \sim \sigma_3$ を演算し(ステップ108)、次いで、次式(数4)により1ワードエラーか否かを判別する(ステップ109)。

【0042】

【数3】

$$\sigma_1 = S1^2 + S0 * S2$$

$$\sigma_2 = S2^2 + S1 * S3$$

10

*クシジョンカウンタ(INSTCNT)をコントロールしている。また、シンドローム演算と訂正処理を同時に行うが、訂正処理されている系列は、同時に行われるシンドローム演算の1系列前のシンドロームであるので、フラグコントローラ(FLGCONT)がシンドローム演算に関する情報およびフラグを記憶し、この情報およびフラグが訂正処理に用いられる。

【0037】次に、図6を参照してメイントラックと補助トラックのC1訂正処理について説明する。C1訂正処理がスタートすると(ステップ101)、先ず、次式(数1)の上段に示す式[C1]によりシンドロームS0～S3をチェックし(ステップ102)、次いで、次式(数2)に示すシンドロームS0～S3を $\alpha \rightarrow i$ 変換してレジスタ8に格納する(ステップ103)。

【0038】

【数1】

$$\sigma_3 = S1 * S2 + S0 * S3$$

30 【0043】

【数4】

$$\sigma_1 + \sigma_2 + \sigma_3 = 0$$

1ワードエラー

$$\sigma_1 + \sigma_2 + \sigma_3 \neq 0$$

1ワードエラー以上

【0044】1ワードエラーの場合には次式(数5)に基づいて1ワード訂正を行って訂正データを書き込み(ステップ110)、次いで表1にも基づいてC1エラーフラグF0に「1」を書き込む(ステップ111)。次いで、ブロックアドレスを1つインクリメントし(ステップ112)、ステップ107に進む。

40 【0045】

【数5】〔1ワード訂正〕

$$\text{エラーの位置: } Xi = S1 / S0$$

$$\text{エラーの値: } Ei = S0$$

$$\text{訂正: } Wi = S0 + Di \quad (Di \dots \text{エラーデータ})$$

【0046】他方、ステップ109において1ワードエラーでない場合には、次式(数6)により2ワードエラーを検出するためのX1、X2、 $\phi_1 \sim \phi_3$ を演算し(ステップ113)、次いで、次式(数7)により2ワードエラーか否かを判別する(ステップ114)。

【0047】

X1~X2, $\phi 1$, $\phi 2$, $\phi 3$ 【数6】

$$X1 = \frac{\sigma 3}{\sigma 1} = \frac{S1 * S2 + S0 * S3}{S1^2 + S0 * S2}$$

$$X2 = \frac{\sigma 2}{\sigma 1} = \frac{S2^2 + S1 * S3}{S1^2 + S0 * S2}$$

$$\phi 1 = S2 + X1 * S1 + X2 * S0$$

$$\phi 2 = S3 + X1 * S2 + X2 * S1$$

$$\phi 3 = S4 + X1 * S3 + X2 * S2 \dots (\text{C2訂正時のみ})$$

【0048】

【数7】

 $\phi 1 + \phi 2 + \phi 3 = 0$ 2ワードエラー $\phi 1 + \phi 2 + \phi 3 \neq 0$ 2ワードエラー以上

【0049】そして、2ワードエラーの場合には次式
 (数8)に基づいて2ワード訂正を行って(ステップ1
 15)次式(数9)により訂正データW_i, W_jを書き

込み(ステップ116)、次いで表1に示すようにC1
 20 エラーフラグF0、F1に「1」を書き込む(ステップ
 117)。次いで、ブロックアドレスを1つインクリメ
 ントし(ステップ118)、ステップ107に進む。

【0050】

【数8】

13

2ワード訂正

2エラーの位置を X_i, X_j とする。

$$X_i + X_j = C_1$$

$$X_i * X_j = C_2$$

として

 X_i, X_j を $f(Y) = 0$ の根として定義すると

$$f(Y) = (Y + X_i)(Y + X_j) = 0$$

$$Y^2 + C_1 * Y + C_2 = 0$$

 C_1, C_2 を $\sigma_1 \sim \sigma_3$ で表すと

$$X_i + X_j = \frac{\sigma_3}{\sigma_1} = X_1 = C_1$$

$$X_i * X_j = \frac{\sigma_2}{\sigma_1} = X_2 = C_2$$

$$f(Y) \rightarrow Y^2 + C_1 * Y + C_2 = 0$$

$$Y = C_1 * Z \text{ として}$$

$$C_1^2 * Z^2 + C_1^2 * Z + C_2 = 0$$

 C_1 で除算して

$$Z^2 + Z + \frac{C_2}{C_1^2} = 0$$

 C_2 / C_1^2 と Z の対照ROMにより Z を求めて

$$X_i = C_1 * Z$$

$$X_j = C_1 + X_i$$

【0051】

【数9】 $[W_i, W_j]$ 訂正

$$S_0 = E_i + E_j$$

$$S_1 = X_i * E_i + X_j * E_j \text{ より}$$

$$X_j * S_0 + S_1 = (X_i + X_j) * E_i$$

$$E_i = (X_j * S_0 + S_1) / C_1$$

$$E_j = S_0 + E_i$$

$$W_i = E_i + D_i$$

$$W_j = E_j + D_j$$

【0052】また、ステップ114において2ワードエラーでない場合には表1に示すように C_1 エラーフラグ F_0, F_1, F_2 に共に「1」を書き込み(ステップ119)、次いで、ブロックアドレスを1つインクリメン

とし(ステップ120)、ステップ107に進む。

40 【0053】次に、図7ないし図9を参照して C_2 訂正処理について説明する。この C_2 訂正処理は C_1 訂正を全ブロック($W=0 \sim 511$)について行った後スタートし(ステップ121)、先ず、上式(数1)の下段 $[C_2]$ によりシンドローム $S_0 \sim S_5$ をチェックし(ステップ122)、次いで、上式(数2)の下段に示すシンドローム $S_0 \sim S_5$ を $\alpha \rightarrow i$ 変換してレジスタ8に格納する(ステップ123)。次いで、 C_1 エラーフラグを読み出して次式(数10)によりエラーフラグの数 $N(E)$ とエラー位置 X_i を検出する(ステップ124)。

【0054】

【数10】 (C1 Flag Calculate)

Read : C1 Flag Location

Count : C1 Flag Number

Register : C1 Flag Location X1, X2, X3,
X4, X5, X6

【0055】ここで、図1～図4において説明したように、ステップ124では $V=0, 1, 48, 49, 96, 97, 144, 145, 192, 193, 240, 241, 288, 289, 336, 337$ のブロックではC1エラーフラグを読み出してC1フラグ評価を行い、そのエラーの位置と数をカウントして記憶するが、他のブロックではC1エラーフラグを読み出さない。続くステップ125では次式(数11)に示すような前演算を行う。

【0056】

$$(X1 + X6)$$

$$* (X2 + X6) (X3 + X6)$$

$$(X1 + X5) (X2 + X5)$$

$$(X1 + X4)$$

【0057】そして、シンδροーム $S0 \sim S5$ が全て「0」か否かを判別することによりエラー数が「0」か否かを判別し(ステップ126)、「0」の場合にはC2エラーフラグ $F0, F1$ に「0」を書き込み(ステップ127)、次いでブロックアドレスを1つインクリメントし(ステップ128)、全ブロック($V=0 \sim 383$)が終了しない場合にはステップ122に戻り、終了した場合にはこのC2訂正処理を終了する(ステップ129)。

【0058】他方、ステップ126においてシンδροーム $S0 \sim S5$ が全て「0」でない場合には、上式(数3)に基づいて1ワードエラーを検出するための変形シンδροーム $\sigma_1 \sim \sigma_3$ を演算し(ステップ131)、次いで、上式(数4)により1ワードエラーか否かを判別する(ステップ132)。そして、1ワードエラーの場合には上式(数5)に基づいて1ワード訂正を行って訂正データを書き込み(ステップ133)、次いでC2エラーフラグ $F0, F1$ に「0」を書き込む(ステップ134)。次いで、ブロックアドレスを1つインクリメントし(ステップ135)、ステップ129に進む。

【0059】他方、ステップ132において1ワードエラーでない場合には、上式(数6)により2ワードエラーを検出するための変形シンδροーム $X1, X2, \phi_1 \sim \phi_3$ を演算し(ステップ136)、次いで、上式(数7)により2ワードエラーか否かを判別する(ステップ137)。

【0060】そして、2ワードエラーの場合には上式(数8)に基づいて2ワード訂正を行って(ステップ138)上式(数9)により訂正データ W_i, W_j を書き込

* 【数11】

$$X1 + X2 = B1$$

$$X1 * X2 = B2$$

$$B1 + X3 = C1$$

$$B1 * X3 + B2 = C2$$

$$B2 * X3 = C3$$

$$C1 + X4 = D1$$

$$C1 * X4 + C2 = D2$$

$$C2 * X4 + C3 = D3$$

$$10 \quad C3 * X4 = D4$$

$$D1 * X5 = E1$$

$$D1 * X5 + D2 = E2$$

$$D2 * X5 + D3 = E3$$

$$D3 * X5 + D4 = E4$$

$$* \quad D4 * X5 = E5$$

$$(X4 + X6) (X5 + X6) = I6$$

$$(X3 + X5) (X4 + X5) = I5$$

$$(X2 + X4) (X3 + X4) = I4$$

$$(X1 + X3) (X2 + X3) = I3$$

$$(X1 + X2) = I2$$

み(ステップ139)、次いでC2エラーフラグ $F0, F1$ に「0」を書き込む(ステップ140)。次いで、ブロックアドレスを1つインクリメントし(ステップ141)、ステップ129に進む。また、ステップ137において2ワードエラーでない場合には図8に示すイレージャルーチンに進む。

【0061】次にイレージャルーチンについて説明する。まず、C1エラーフラグ $F1$ の数が「0」か否かを判別し(ステップ144)、NOの場合にはC1エラーフラグ $F1$ の数が5以下か否かを判別し(ステップ145)、5以下の場合には5か否かを判別する(ステップ146)。そして、C1エラーフラグ $F1$ の数が5でない場合には図9に詳しく示すシンδροーム修正ルーチンを実行し、他方、5の場合には次式により $N=5 \sim 1$ のイレージャを実行し(ステップ147)、次いでブロックアドレスを1つインクリメントし(ステップ148)、ステップ121に戻る。

【0062】また、ステップ144においてC1エラーフラグ $F1$ の数が「0」の場合には、C2エラーフラグ $F0$ に「1」を書き込み(ステップ149)、次いでブロックアドレスを1つインクリメントし(ステップ147)、ステップ121に戻る。また、ステップ145においてC1エラーフラグ $F1$ の数が5以下でない場合にはステップ152以下に分岐する。

【0063】ステップ152においてC1エラーフラグ $F2$ の数が「0」の場合には、C2エラーフラグ $F1$ に「1」を書き込み(ステップ153)、次いでブロックアドレスを1つインクリメントし(ステップ154)、ステップ121に戻る。また、ステップ152において

17

C1エラーフラグF2の数が3以下の場合には図9に詳しく示すシンドローム修正ルーチンを実行し、また、ステップ156においてC2エラーフラグF2の数が5以下の場合には次式に示す $N=5\sim 1$ のNイレージャを実行し、次いでブロックアドレスを1つインクリメントし（ステップ158）、ステップ121に戻る。

【0064】また、ステップ159においてC2エラーフラグF2の数が6でない場合にはC2エラーフラグF1に「1」を書き込み（ステップ160）、次いでブロックアドレスを1つインクリメントし（ステップ161）、ステップ121に戻る。また、ステップ162においてC2エラーフラグF2の数が6の場合には次式により $N=6\sim 1$ のNイレージャを実行し（ステップ163）、次いでブロックアドレスを1つインクリメントし（ステップ164）、ステップ121に戻る。なお、図*

$$T4 = S4 + D1 * S3 + D2 * S2 + D3 * S1 + D4 * S0$$

$$Y5 = T4 / I5$$

〔シンドローム修正〕

$$S0 + Y5 \rightarrow S0$$

$$S1 + Y5 * X5 \rightarrow S1$$

$$S2 + Y5 * X5^2 \rightarrow S2$$

$$S3 + Y5 * X5^3 \rightarrow S3$$

【0067】

【数14】〔4Erasure, Y4〕

$$T3 = S3 + C1 * S2 + C2 * S1 + C3 * S0$$

$$Y4 = T3 / I4$$

〔シンドローム修正〕

$$S0 + Y4 \rightarrow S0$$

$$S1 + Y4 * X4 \rightarrow S1$$

$$S2 + Y4 * X4^2 \rightarrow S2$$

【0068】

【数15】〔3Erasure, Y3〕

$$T2 = S2 + B1 * S1 + B2 * S0$$

$$Y3 = T2 / I3$$

〔シンドローム修正〕

$$S0 + Y3 \rightarrow S0$$

$$S1 + Y3 * X3 \rightarrow S1$$

【0069】

【数16】〔2Erasure, Y2〕

〔1Erasure, Y1〕

$$T1 = S1 + X1 * S0$$

$$Y2 = T1 / I2$$

$$Y1 = S0 + Y2$$

【0070】上記第1実施例において、C2系列の1個又は2個の誤りを検出するに際し、可能性として、例えば実際は3個ある誤りを1個の誤りと誤認し、あるいは例えば実際は4個ある誤りを2個の誤りと誤認することもありえる。かかる不具合を解消したのが次に説明する第2実施例である。

【0071】この第2実施例は、上記第1実施例とハ

18

*9に示すシンドローム修正ルーチンについては詳細な説明を省略する。

【0065】

【数12】〔6Erasure, Y6〕

$$T5 = S5 + E1 * S4 + E2 * S3 + E3 * S2$$

$$+ E4 * S1 + E5 * S0$$

$$Y6 = T5 / I6$$

〔シンドローム修正〕

$$S0 + Y6 \rightarrow S0$$

$$S1 + Y6 * X6 \rightarrow S1$$

$$S2 + Y6 * X6^2 \rightarrow S2$$

$$S3 + Y6 * X6^3 \rightarrow S3$$

$$S4 + Y6 * X6^4 \rightarrow S4$$

【0066】

【数13】〔5Erasure, Y5〕

ドウェア的には同一であり、エラー訂正の基本的手法は同一である。したがって、第1実施例と異なる点を中心に説明する。第2実施例においては、C1系列は図6と同様の手順で訂正が行われるが、C2系列の訂正手順において異なる部分がある。

【0072】図10はリードソロモン符号のC2系列を訂正する第2実施例のルーチンを説明するフローチャートである。このフローは図7（第1実施例）に対応するものであり、図7と異なるところのみ説明する。図10中、ステップ150、152、156、158は図7のフローに対して新たに加えられたステップである。ステップ150、158は共に訂正を行うべきか否かを判断する訂正要否判断のステップである。その詳細はそれぞれ図11と図12に示されている。

【0073】図10～図12を参照して第2実施例の動作について説明する。図10のステップ133にて1ワード訂正のための演算を行った後、ステップ150に入り、図11に示されるように、ステップ160にて先のステップ124で読み込んだC1系列のフラグ中、訂正しようとする1つのワードのF2フラグについてチェックし、F2=1であるか否かを判断する。F2=1であればステップ134へ行き訂正を実行し、一方、F2=0であればステップ162でC1系列のフラグ中、訂正しようとする1つのワードのF1フラグについてチェックし、F1=1なら訂正を実行し、F1=0なら訂正は実行せずステップ152にてフラグF0、F1、F2をそれぞれ1に書き替えて、ステップ135へ行く。ステップ134で訂正を実行した場合も、ステップ134AでフラグF0、F1、F2を0としてからステップ135へ行く。C1系列のエラーフラグF2の値のチェックはこのように訂正しようとするワードについて行えばよいのでこのプロセスにより高速化が妨げられるという不都合が生じることはない。

【0074】2ワードエラー訂正のルーチンについても、上記と同様であり、ステップ156がステップ150に対応し、ステップ158がステップ152に対応している。図12は図10の訂正要否判断ステップ156の内容を示したフローチャートであり、ここでは図11同様の手順が行われる。すなわちステップ170、172にステップ160、162に各々対応している。

【0075】上記の動作以外は図7と同様である。このように本発明の第2実施例ではC2系列の訂正演算により得られた誤りデータの位置を基に、その誤りデータのC1フラグを読み込み、そのC1系列のエラーフラグの値によってC2系列の訂正動作を実行するか否かを判断している。この第2実施例は上記第1実施例とハードウェア的に同一のもので説明したが、これに限るものではなく、C2系列エラー訂正手段において、前記C2系列の訂正演算により得られた誤りデータの位置を基に、その誤りデータのC1フラグを読み込み、そのC1系列のエラーフラグの値によって前記C2系列の訂正動作を実行するか否かを判断する手段を備えていればよい。

【0076】

【発明の効果】以上説明したように本発明によれば、1フレームのデータがsワード×Wブロックで構成されたC1系列と、1フレームのデータがtワード×Vブロックで構成されたC2系列の2重のリードソロモン符号を用いてC2系列をエラー訂正する場合に、C1系列のワードとC2系列のワードの巡回性に基づいてC2系列のブロックをC1系列のワードが重複するグループにグループ分けすることによりC1系列のエラー訂正時のエラーフラグがグループ毎に1回読み出すので、グループ分けされたC2系列の他のブロックではC1系列のエラー訂正時のエラーフラグを読み出す必要がなくなり、したがって、高速でエラー訂正することができる。

【0077】また、本発明の他の態様ではC2系列の訂正演算により得られた誤りデータの位置を基に、その誤りデータのC1フラグを読み込み、そのC1系列の各ワードのエラーフラグの値によってC2系列の訂正動作を実行するか否かを判断するようにしているので、例えば3個の誤りを1個と誤認したり、4個の誤りを2個の誤

りと誤認するという不都合がなくなり、全体として訂正誤りの生ずる確率が大幅に低下する。

【図面の簡単な説明】

【図1】本発明に係るエラー訂正装置の第1及び第2実施例におけるエラー訂正方法を示す説明図である。

【図2】C1系列のワードとC2系列のワードの関係を示す説明図である。

【図3】C2系列の第0～第95系列とC1系列の関係を示す説明図である。

10 【図4】C2系列の第96～第191系列とC1系列の関係を示す説明図である。

【図5】本発明に係る誤り訂正装置の第1及び第2実施例を示すブロック図である。

【図6】リードソロモン符号のC1系列を訂正する第1及び第2実施例のルーチンを説明するためのフローチャートである。

【図7】リードソロモン符号のC2系列を訂正する第1実施例のルーチンを説明するためのフローチャートである。

20 【図8】図7のC2系列訂正ルーチンのイレージャルーチンを説明するためのフローチャートである。

【図9】シンドローム修正ルーチンを説明するためのフローチャートである。

【図10】リードソロモン符号のC2系列を訂正する第2実施例のルーチンを説明するためのフローチャートである。

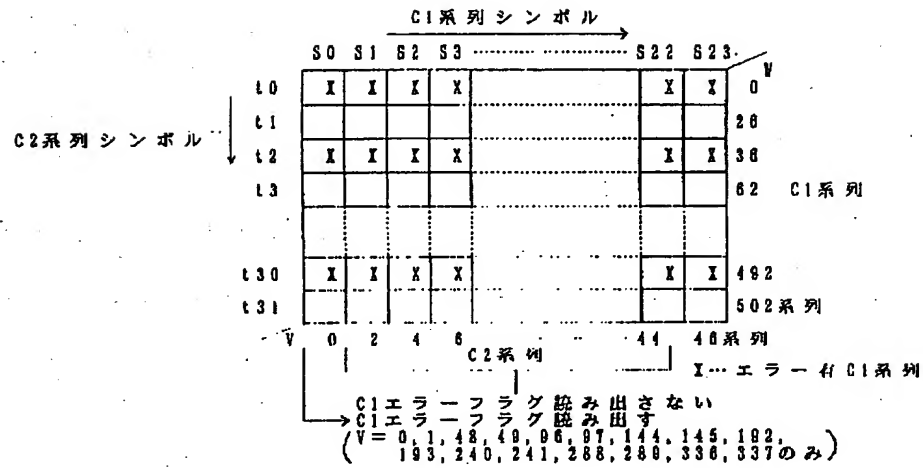
【図11】図10に示したステップ150を詳細に示すフローチャートである。

30 【図12】図10に示したステップ156を詳細に示すフローチャートである。

【符号の説明】

- 2 フラグロケーション回路
- 22 シンドローム演算ブロック
- 23 ラッチブロック
- 24 アドレスブロック
- 25 インストラクションブロック
- 26 訂正ブロック

【図1】



【図2】

| シンボル | C2ワード: V | | | | | | | | C1ワード |
|------|----------|-----|-----|-----|--|-----|-----|-----|-------|
| t | 0 | 2 | 4 | 6 | | 42 | 44 | 46 | V |
| 0 | S0 | S1 | S2 | S3 | | S21 | S22 | S23 | 0 |
| 1 | S0 | S1 | S2 | S3 | | S21 | S22 | S23 | 20 |
| 2 | S0 | S1 | S2 | S3 | | S21 | S22 | S23 | 36 |
| 3 | S0 | S1 | S2 | S3 | | S21 | S22 | S23 | 62 |
| 4 | S0 | S1 | S2 | S3 | | S21 | S22 | S23 | 172 |
| 5 | S0 | S1 | S2 | S3 | | S21 | S22 | S23 | 482 |
| ... | ... | ... | ... | ... | | ... | ... | ... | ... |
| 30 | S0 | S1 | S2 | S3 | | S21 | S22 | S23 | 482 |
| 31 | S0 | S1 | S2 | S3 | | S21 | S22 | S23 | 502 |

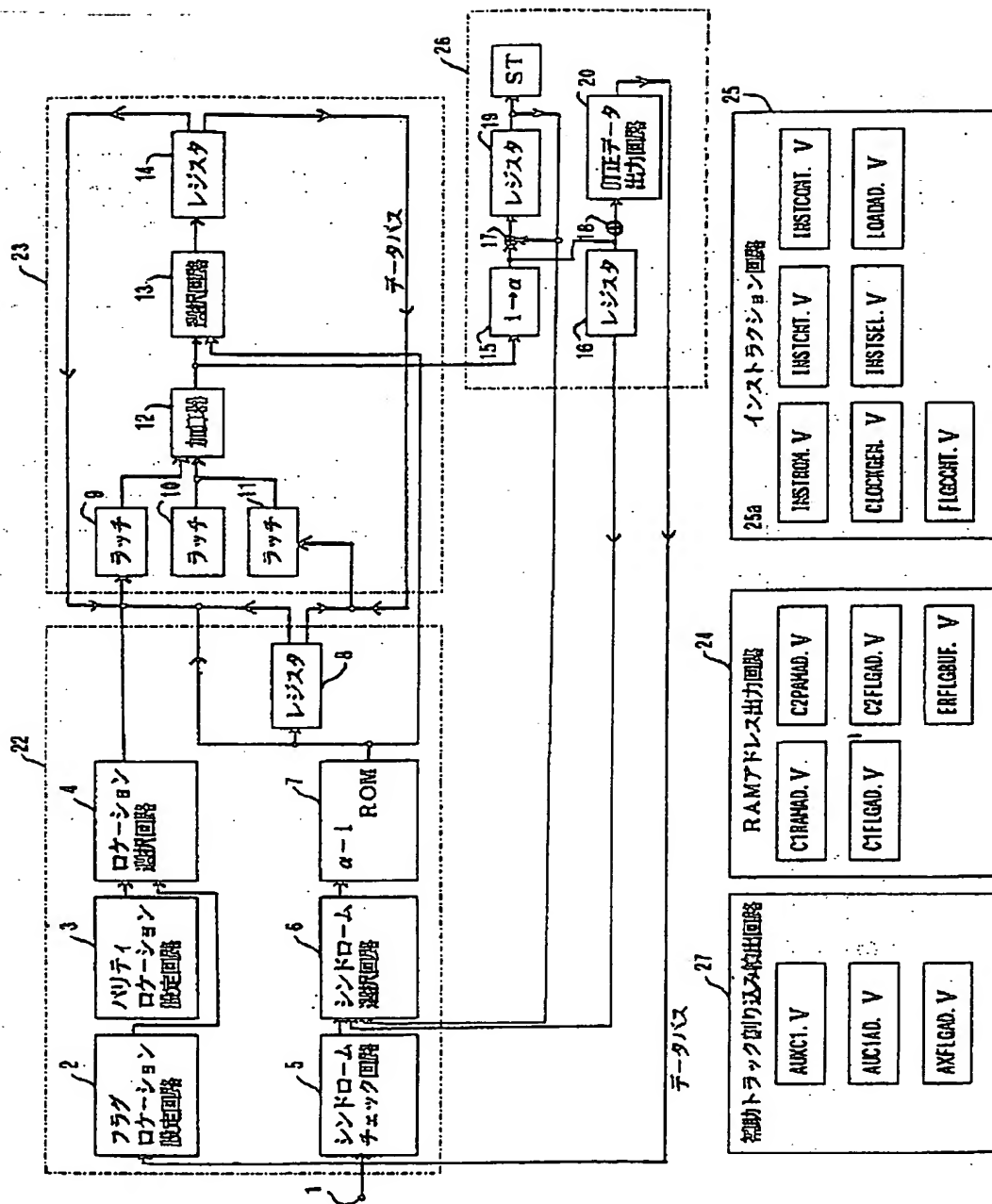
【図3】

| C 2 7 - R : V | | | | |
|---------------|------------------|------------------|-----------------|-----------------|
| シンボル | 0, 2, 4...44, 46 | 1, 3, 5...45, 47 | 48, 50...92, 94 | 49, 51...93, 95 |
| t | | | | |
| 0 | | | | |
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |
| 5 | | | | |
| 6 | | | | |
| 7 | | | | |
| 8 | | | | |
| 9 | | | | |
| 10 | | | | |
| 11 | | | | |
| 12 | | | | |
| 13 | | | | |
| 14 | | | | |
| 15 | | | | |
| 16 | | | | |
| 17 | | | | |
| 18 | | | | |
| 19 | | | | |
| 20 | | | | |
| 21 | | | | |
| 22 | | | | |
| 23 | | | | |
| 24 | | | | |
| 25 | | | | |
| 26 | | | | |
| 27 | | | | |
| 28 | | | | |
| 29 | | | | |
| 30 | | | | |
| 31 | | | | |

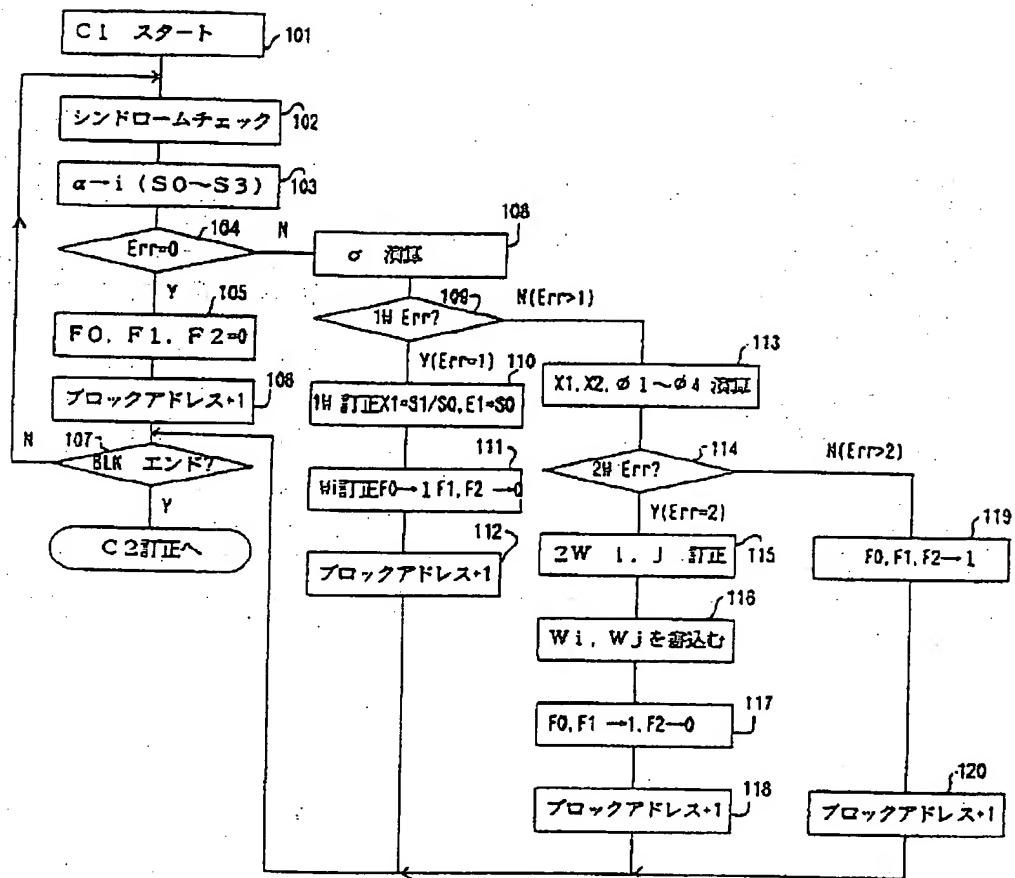
【図4】

| シンボル | C2ワード: V | | | |
|------|--------------------|--------------------|------------------------|----------------------|
| | 96, 98... 140, 142 | 97, 99... 141, 143 | 144, 146, ... 188, 190 | 145, 147... 189, 191 |
| 0 | W=32 | W=33 | W=48 | W=49 |
| 1 | W=58 | W=59 | W=74 | W=75 |
| 2 | W=68 | W=69 | W=84 | W=85 |
| 3 | W=94 | W=95 | W=110 | W=111 |
| 4 | W=104 | W=105 | W=120 | W=121 |
| 5 | W=114 | W=115 | W=2 | W=3 |
| 6 | W=12 | W=13 | W=28 | W=29 |
| 7 | W=22 | W=23 | W=38 | W=39 |
| 8 | W=166 | W=167 | W=176 | W=177 |
| 9 | W=186 | W=187 | W=202 | W=203 |
| 10 | W=196 | W=197 | W=212 | W=213 |
| 11 | W=222 | W=223 | W=238 | W=239 |
| 12 | W=232 | W=233 | W=248 | W=249 |
| 13 | W=242 | W=243 | W=130 | W=131 |
| 14 | W=140 | W=141 | W=156 | W=157 |
| 15 | W=150 | W=151 | W=166 | W=167 |
| 16 | W=288 | W=289 | W=304 | W=305 |
| 17 | W=314 | W=315 | W=330 | W=331 |
| 18 | W=324 | W=325 | W=340 | W=341 |
| 19 | W=350 | W=351 | W=366 | W=367 |
| 20 | W=360 | W=361 | W=376 | W=377 |
| 21 | W=370 | W=371 | W=258 | W=259 |
| 22 | W=268 | W=269 | W=284 | W=285 |
| 23 | W=278 | W=279 | W=294 | W=295 |
| 24 | W=416 | W=417 | W=432 | W=433 |
| 25 | W=422 | W=423 | W=458 | W=459 |
| 26 | W=452 | W=453 | W=468 | W=469 |
| 27 | W=478 | W=479 | W=494 | W=495 |
| 28 | W=488 | W=489 | W=504 | W=505 |
| 29 | W=498 | W=499 | W=386 | W=387 |
| 30 | W=396 | W=397 | W=412 | W=413 |
| 31 | W=406 | W=407 | W=422 | W=423 |

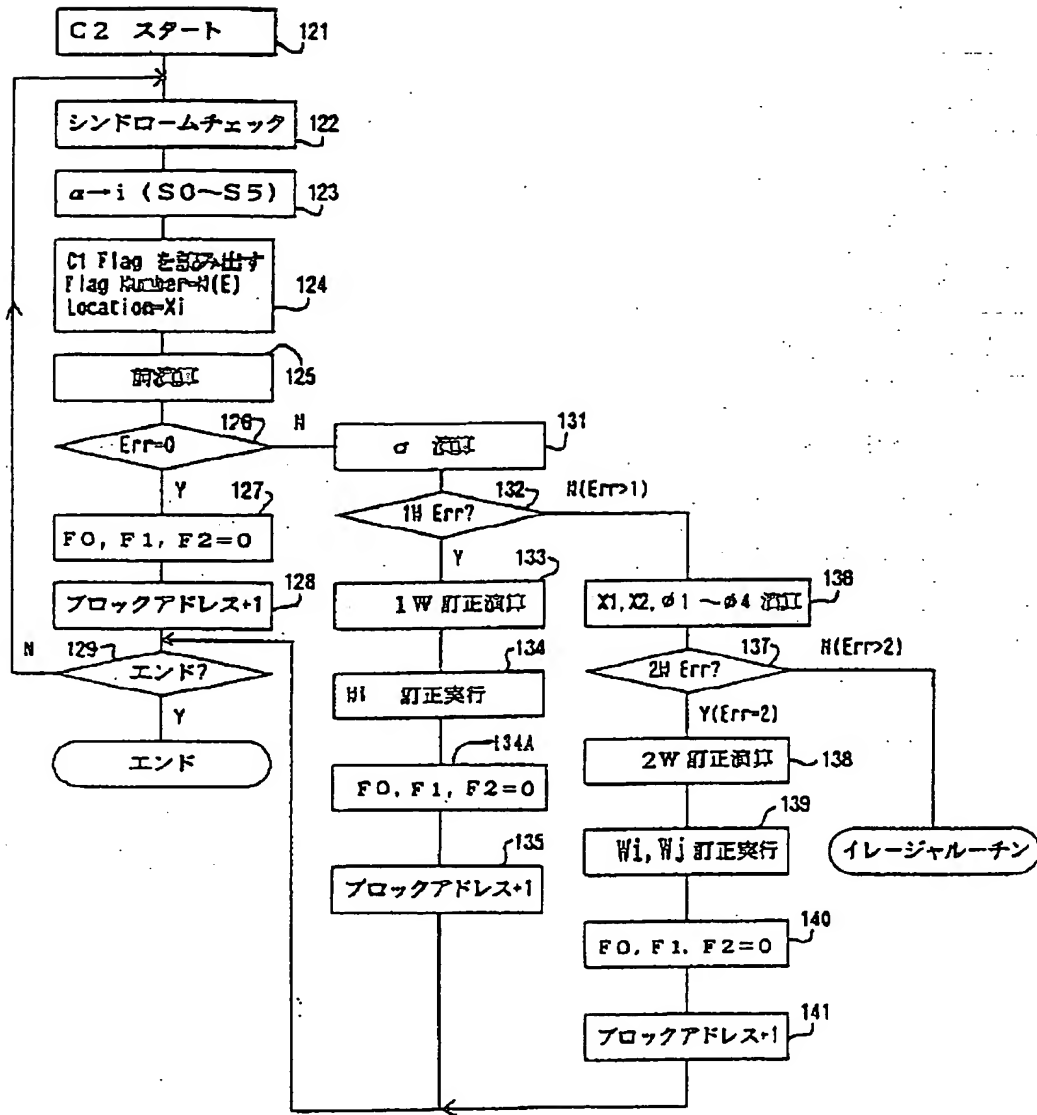
- 15 -



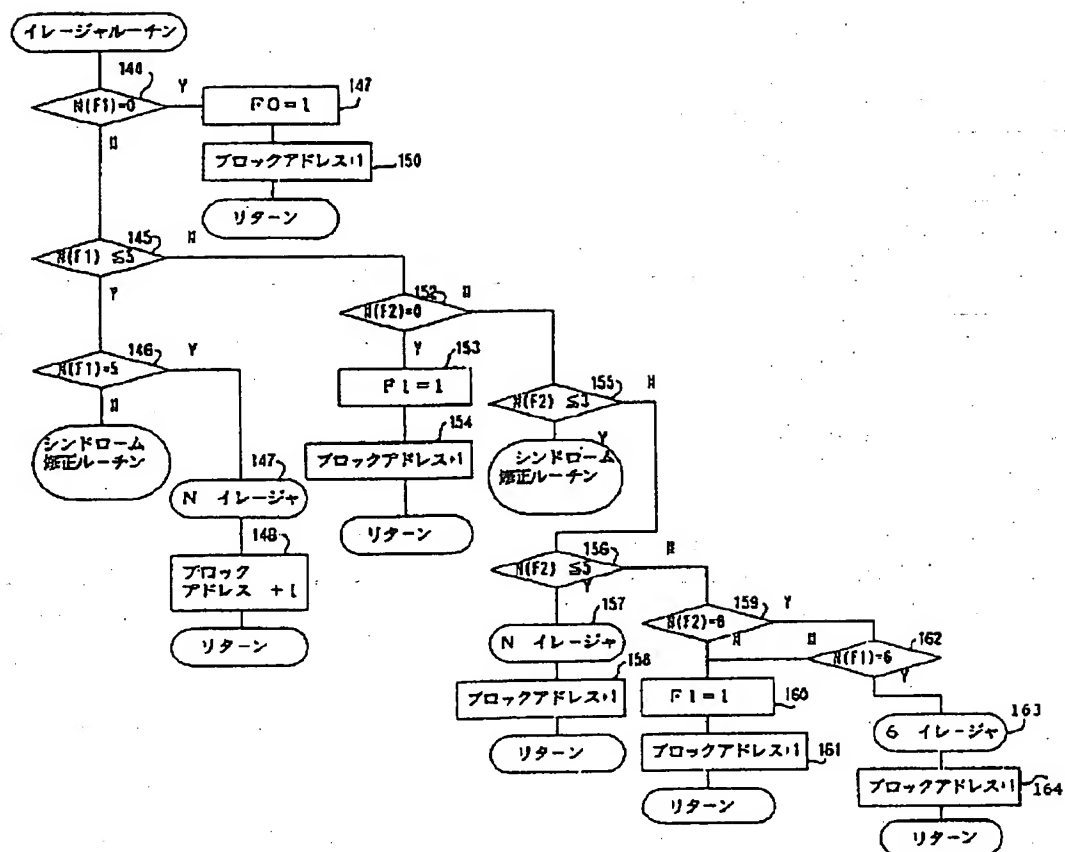
【図6】



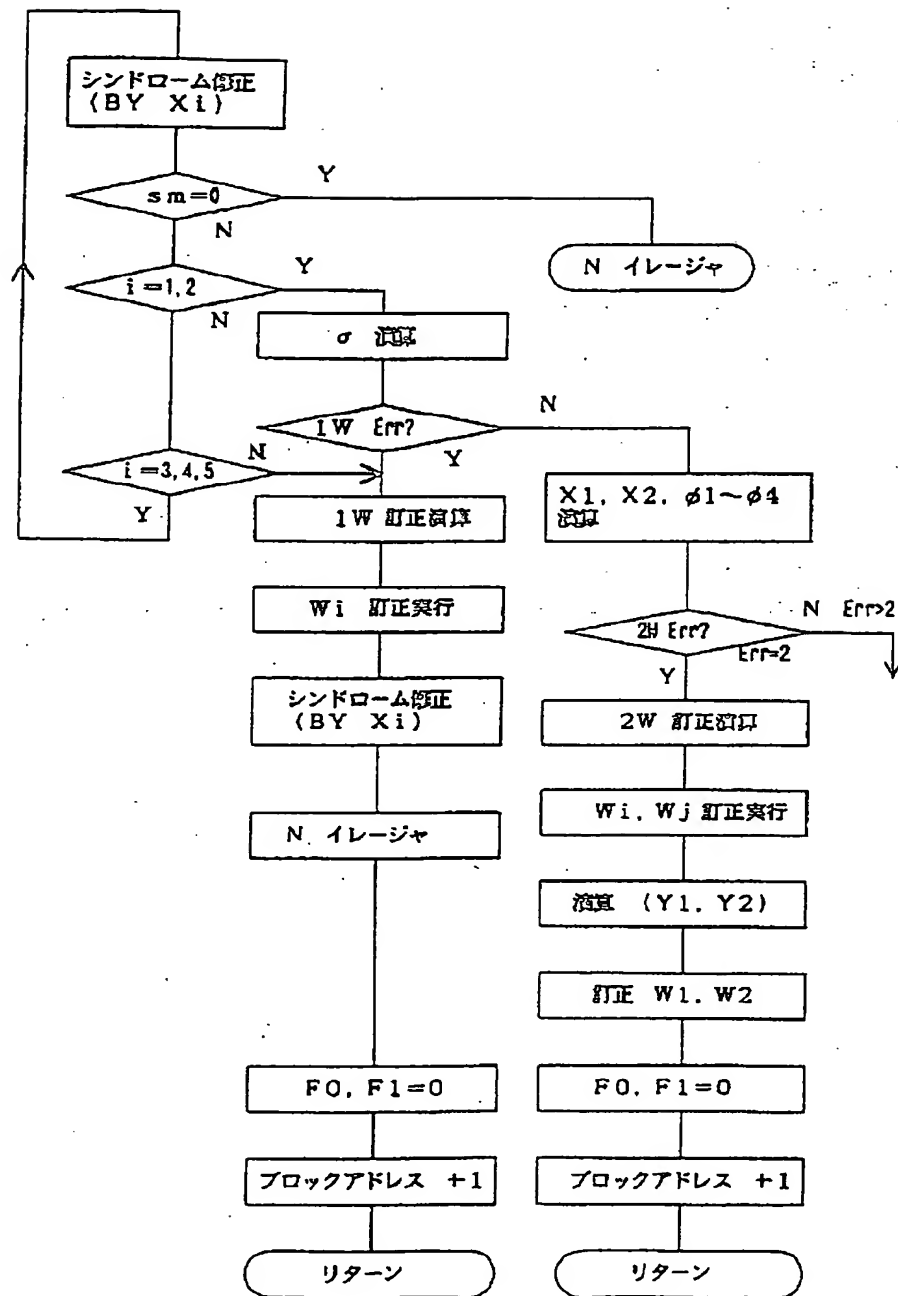
【図7】



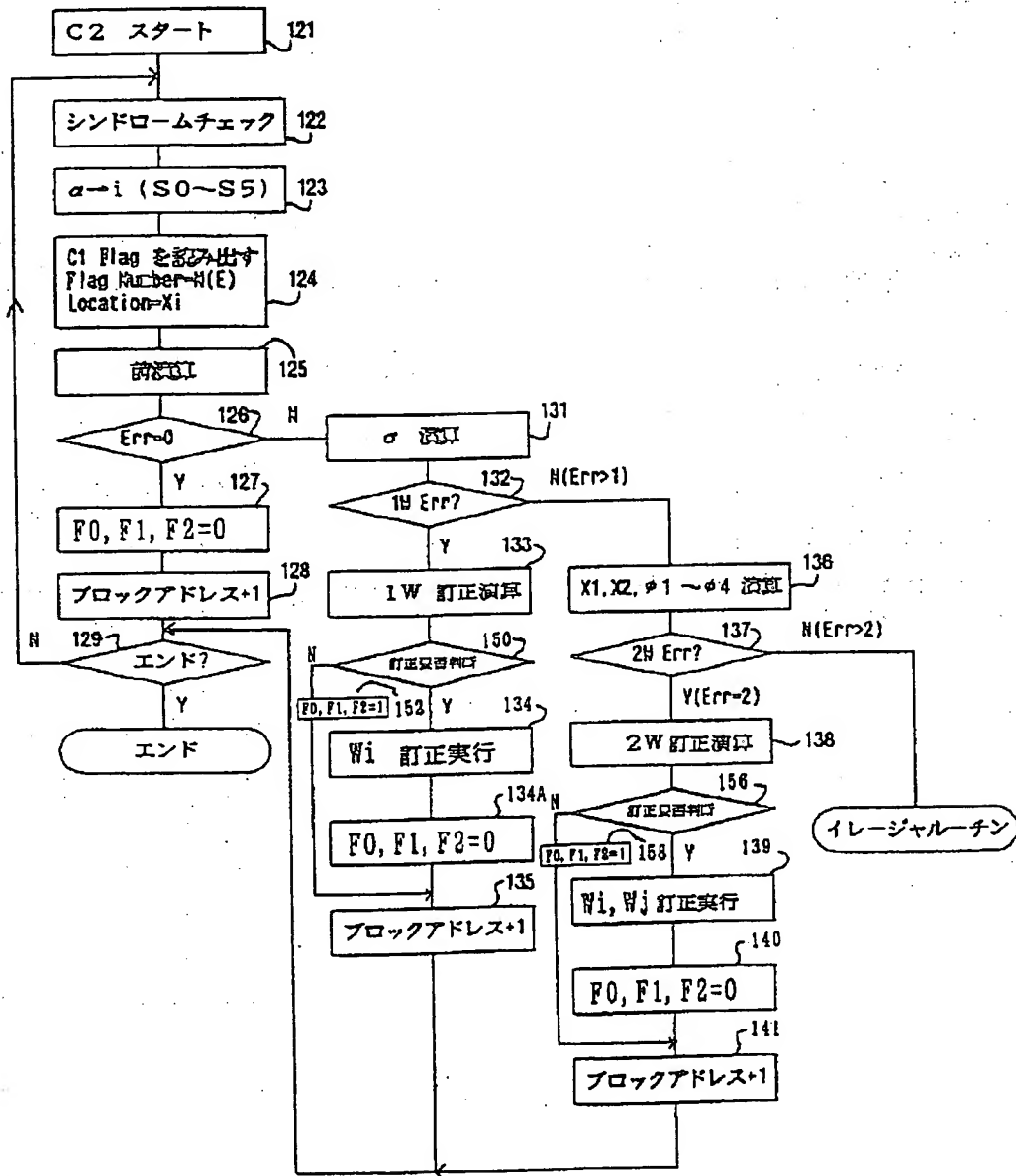
【図 8】



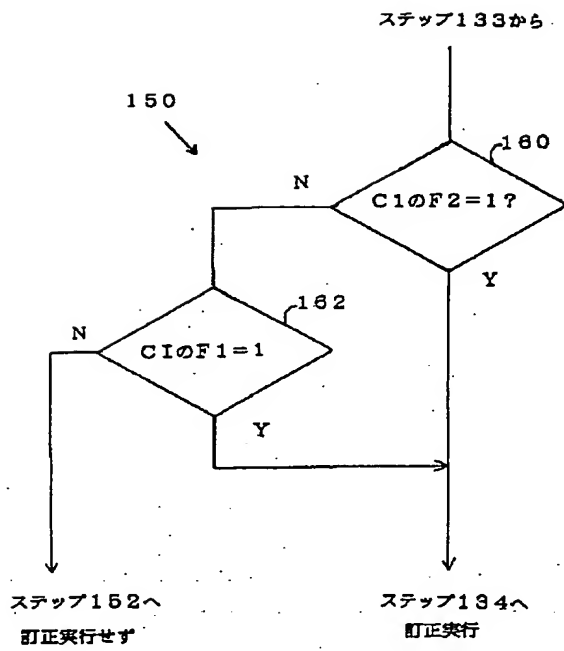
【図9】



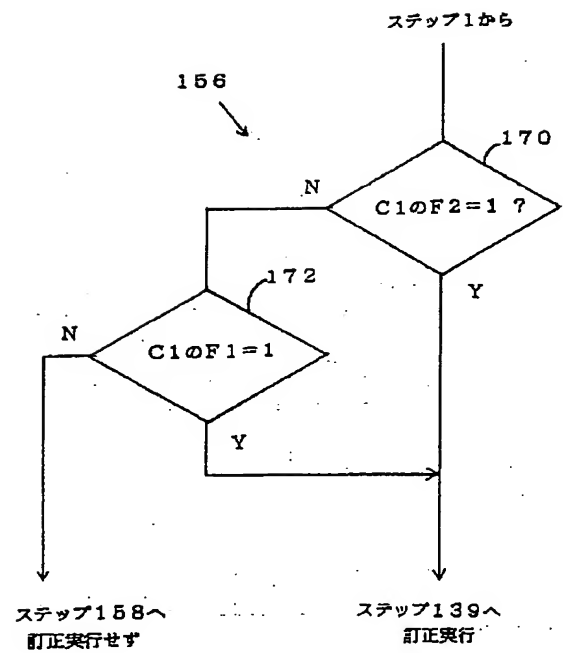
【図10】



【図11】



【図12】



THIS PAGE BLANK (USPTO)